Ultimate Python Cheat Sheet

Originalartikel

Backup

<html> <div class=„crayons-articlemain crayons-articlebody text-styles specbody“> In this post you will find a very handy Python memory aid Dont miss my next python post: Follow @EricTheCoder_
 <hr />
Here is my cheat sheet I created along my learning journey. If you have any recommendations (addition/subtraction) let me know. Naming conventions
 <div class=„highlight js-code-highlight“><pre># Variable lower_snakefirst_name = 'Mike'# Class and module CamelCaseclass InvoiceDetail:# ConstantMAX_USER = 100 # All uppercase# Indentation : 4 spacesif num > 9: print('Small number')</pre> Enter fullscreen modeExit fullscreen mode Data type
 <div class=„highlight js-code-highlight“><pre>name = 'Mike' # stringage = 42 # intprice = 199.99 # floatis_active = True # booleancolors = ['red', 'green', 'blue'] # liststates = ('inactive', 'active', 'archive') # tupleproducts = { 'name': 'iPad Pro', 'price': 199.99 } # dict</pre> Enter fullscreen modeExit fullscreen mode Type conversion
 <div class=„highlight js-code-highlight“><pre># Python is a strong type languagenumber = 50 + „50“ # TypeError# Convert to stringnumber = 50 + int(„50“) # 100# Convert to stringmy_text = str(199.99) # „199.99“# Convert to numbermy_number = int(21.99) # 21my_number = float('21.99') # 21.99# Get typetype(my_text) # <class 'str'>type(my_number) # <class 'float'># Check if number 0 to 9isdigit('8') # True# Check typeisinstance(my_number, int) # True</pre> Enter fullscreen modeExit fullscreen mode Strings
 <div class=„highlight js-code-highlight“><pre>name = 'Mike'# orname = „Mike“# ormessage = „“„This is multilinestring that is easier toread and assign“„“# escape characters \n will do a line breakmessage = „Hello \nWorld“# raw string (ignore escape characters)message = r„https:\\example.com\index.html“# Convert to lower casename.lower() # mike# Convert to upper casename.upper() # MIKE# Convert first char to Capital lettername.capitalize() # Mike# Convert first char of all words to Capital lettername = 'mike taylor'name.title() # Mike Taylor# Chain methodsname = 'MIKE'name.lower().capitalize() # Mikename = 'Mike'# Start with ?name.startswith('M') # true# End with ?name.endswith('ke') # true# String lengthlen(name) # 4# String concatenationfull_name = first_name + ' ' + last_name# String formatfull_name = f„{first_name}{last_name}“# String format number (2 decimal point)number = 12.9997number_string = „{:.2f}“.format(number) # '12.99'# Remove leading and trailing characters (like space or \n)text = ' this is a text with white space 'text.strip() # 'this is a test with white space'name = 'Mike'# Get string first charactername[0] # M# Get string last charactername[-1] # e# Get partial stringname[1:3] # ik# Replacename.replace('M', 'P') # Pike# Find (return pos or -1 if not found)name.find('k') # 2# List to stringcolors = ['red', 'green', 'blue']', '.join(colors) # 'red, green, blue'</pre> Enter fullscreen modeExit fullscreen mode Commons fonctions
 <div class=„highlight js-code-highlight“><pre># Print to consoleprint('Hello World')# Print multiple stringprint('Hello', 'World') # Hello World# Multiple printprint(10 * '-') # ———-# Variable pretty printer (for debug)from pprint import pprintpprint(products) # will output var with formatting# Get keyboard inputname = input('What is your name? ')# Random (between 0 and 1)from random import random print(random()) # 0.26230234411558273# Random beween x and yfrom random

import randintprint(randint(3, 9)) # 5# Roundingnumber = 4.5print(round(number)) # 5number = 4.5163print(round(number, 2)) # 4.52# Pathimport oscurrent_file_path = filefolder_name = os.path.dirname(current_file_path)new_path = os.path.join(folder_name, 'new folder')# Round numbersolution = round(12.9582, 2) # 12.96</pre> Enter fullscreen modeExit fullscreen mode Conditionals
 <div class=„highlight js-code-highlight“><pre>if x == 4: print('x is 4')elif x != 5 and x < 11: print('x is between 6 and 10')else: print('x is 5 or greater than 10')#In or not incolors = ['red', 'green', 'blue', 'yellow']if 'blue' in colors:if 'white' not in colors:# Ternaryprint('y = 10') if y == 10 else print('y != 10') # ShortHand Ternaryis_valid = 'Valid'msg = is_valid or „Not valid“ # 'Valid'# FalsyFalse, None, 0, empty string „“, empty list [], (), {}# TruthyTrue, not zero and not empty value </pre> Enter fullscreen modeExit fullscreen mode Interations
 <div class=„highlight js-code-highlight“><pre># iterating over a sequence (list, string, etc.)for item in items: print(item)# With indexfor index, item in enumerate(items): print(index, item)# Rangefor i in range(10): #0..9 print(i)for i in range(5, 10): #5..9 print(i)# While loopwhile x > 10: print(x) # exit loop if x == 5: break # Jump to next while if x == 3: continue x += 1# For loop dicfor key, value in my_dict.items(): print(key, value)# List comprehension: # values = [(expression) for (value) in (collection)]items = [value*2 for value in items]# List comprehension filtering# values = [expression for value in collection if condition]even_squares = [x * x for x in range(10) if x % 2 == 0]</pre> Enter fullscreen modeExit fullscreen mode List and Tuple
 <div class=„highlight js-code-highlight“><pre># Create a listfruits = ['orange', 'apple', 'melon']# Append to Listfruits.append('banana')# List lengthnb_items = len(fruits)# Remove from listdel fruits[1] #remove apple# List accessfruits[0] # first itemfruits[-1] # last item# Slice my_list[start:finish:step] ([::-1] reverse list) fruits = fruits[1:3]fruits[:3] # first 3fruits[2:] # last 2copy_fruits = fruits[:] # copy# List lengthnb_entry = len(fruits) #Create list from stringcolors = 'red, green, blue'.split(', ')# Array concactcolor1 = ['red', 'blue']color2 = ['green', 'yellow']color3 = color1 + color2# Concat by unpackingcolor3 = [*color1, *color2]# Multiple assignmentname, price = ['iPhone', 599]#Create a Tuple (kind of read only list)colors = ('red', 'green', 'blue')# Sortcolors.sort() # ['blue', 'green', 'red']colors.sort(reverse=True) # ['red', 'green', 'blue']colors.sort(key=lambda color: len(color)) # ['red', 'blue', 'green']</pre> Enter fullscreen modeExit fullscreen mode Dictionaries
 <div class=„highlight js-code-highlight“><pre># Create a empty dictproduct = {}#Create a dict with key/valueproduct = {'id': 100, 'name': 'iPadPro'}# Access dict value by keyprint(product['name']) # iPadPro# Access dictproduct.get('name') # if key not exist return Noneproduct.get('name', 'default value') # if key not exist return default value# Adding a new key/valueproduct['description'] = „Modern mobile device“# Get dict keysproduct.keys() # ['id', 'name', 'description']# Get dic valuesproduct.values() # ['100', 'iPadPro', 'Modern mobile device']# Create a list of dictproducts = [{'id': 100, 'name': 'iPadPro'}, {'id': 200, 'name': 'iPhone 12'}, {'id': 300, 'name': 'Charger'},]# Access list of dictprint(products[2]['name']) # Charger# Search list dictitems_match = [item for product in products if product['id'] == 300]# [{'id': 300, 'name': 'Charger'}]# Sum list dicttotal = sum([product['price'] for product in products])</pre> Enter fullscreen modeExit fullscreen mode Functions
 <div class=„highlight js-code-highlight“><pre># Create a functiondef say_hello(): print('Hello World')# Function with argument (with default value)def say_hello(name = 'no name'): print(f„Hello {name}“) # Function with argument (with optional value)def say_hello(name = None): if name: print(f„Hello {name}“) else: print('Hello World')# Call a functionsay_hello('Mike') # Hello Mike# Call using keyword argumentsay_hello(name = 'Mike') # Function returning a valuedef add(num1, num2): return num1 + num2num = add(10, 20) # 30# Arbitrary numbers of arguments *argsdef say_hello(*names): for name in names: print(f„Hello {name}“)# Arbitrary numbers of keywords arguments **kwargsdef say_hello(**kwargs): print(kwargs['name']) print(kwargs['age'])say_hello(name = 'Mike', age = 45)# Lambda functionx = lambda num : num + 10print(x(20)) # 30</pre> Enter fullscreen modeExit fullscreen mode Date and time
 <div class=„highlight

js-code-highlight“><pre>from datetime import datetime, timedelta# Return the current date and time.datetime.now()# Create a date time objectdate = datetime(2020,12,31) # Dec 31 2020# Add to date/time (weeks, days, hours, minutes, seconds) new_year = date + timedelta(days=1) # Jan 1 2021# Format a date to stringnew_year.strftime('%Y/%m/%d %H %M %S') # 2021/01/01 00 00 00 new_year.strftime('%A, %b %d') # Friday, Jan 01# Extract from dateyear = new_year.year # 2021month = new_year.month # 01</pre> Enter fullscreen modeExit fullscreen mode File
 <div class=„highlight js-code-highlight“><pre># Reading a file and storing its linesfilename = 'demo.txt'with open(filename) as file: lines = file.readlines()for line in lines: print(line)# Writing to a filefilename = 'settings.txt'with open(filename, 'w') as file: file.write(„MAX_USER = 100“)# File exist?from os import pathpath.exists('templates/index.html') # True/False# CSVimport csvcsv_file = 'export.csv'csv_columns = products[0].keys() # ['id', 'name']with open(csv_file, 'w') as csvfile: writer = csv.DictWriter(csvfile, fieldnames=csv_columns) writer.writeheader() for item in products: writer.writerow(item)</pre> Enter fullscreen modeExit fullscreen mode <div class=„highlight js-code-highlight“><pre>age_string = input('Your age? ')try: age = int(age_string)except ValueError: print(„Please enter a numeric value“)else: print(„Your age is saved!“)</pre> Enter fullscreen modeExit fullscreen mode OOP
 <div class=„highlight js-code-highlight“><pre># Create a classclass Product: pass# Class attributeclass Product: nb_products = 0print(Product.nb_products) # 0# Create new object instanceproduct_1 = Product()# Object instance attributesclass Product: def init(self, name, price): self.name = name self.price = price# Create instance with attributesproduct_1 = Product('iPadPro', 699.99)product_2 = Product('iPhone12', 799.99)print(product_1.name) # iPadPro# instance methodclass Product() def display_price(self): return f„Price : {self.price}„print(product_1.display_price())# class methodclass Product: # ... @classmethod def create_default(cls): # create a instance return cls('Product', 0) # default name, default priceproduct_3 = Product.create_default() # static methodclass Product: # ... @staticmethod def trunc_text(word, nb_char): return word[:nb_char] + '...' product_3 = Product.trunc_text('This is a blog', 5) # This i... # Python Inheritanceclass WebProduct(Product): def init(self, name, price, web_code): super().init(name, price) self.web_code = web_code# Private scope (naming convention only)def init(self, price): self.price = price# Getter and setterclass Product: def init(self): self.price = 0 @property def price(self): return self.price @price.setter def price(self, value): self.price = value# Mixinsclass Mixin1(object): def test(self): print „Mixin1“class Mixin2(object): def test(self): print „Mixin2“class MyClass(Mixin2, Mixin1, BaseClass): passobj = MyClass()obj.test() # Mixin2</pre> Enter fullscreen modeExit fullscreen mode <div class=„highlight js-code-highlight“><pre>from my_module import my_function# import local file (same folder)# my_module.py will be the file name# my_function is a function inside my_module.py</pre> Enter fullscreen modeExit fullscreen mode Dont miss my next python post: Follow @EricTheCoder_ <hr /> Here is my cheat sheet I created along my learning journey. If you have any recommendations (addition/subtraction) let me know. </html>